Building A Highly Accurate Mandarin Speech Recognizer With Language-Independent Technologies and Language-Dependent Modules

Mei-Yuh Hwang, *Member, IEEE*, Gang Peng, Mari Ostendorf, *Fellow, IEEE*, Wen Wang, *Member, IEEE*, Arlo Faria, *Member, IEEE*, and Aaron Heidel

***Abstract*—We describe a system for highly accurate large-vocabulary Mandarin speech recognition. The prevailing hidden Markov model based technologies are essentially language independent and constitute the backbone of our system. These include minimum-phone-error discriminative training and maximum-likelihood linear regression adaptation, among others. Additionally, careful considerations are taken into account for Mandarin-specific issues including lexical word segmentation, tone modeling, phone set design, and automatic acoustic segmentation. Our system comprises two sets of acoustic models for the purposes of cross adaptation. The systems are designed to be complementary in terms of errors but with similar overall accuracy by using different phone sets and different combinations of discriminative learning. The outputs of the two subsystems are then rescored by an adapted n-gram language model. Final confusion network combination yielded 9.1% character error rate on the DARPA GALE 2007 official evaluation, the best Mandarin recognition system in that year.**

***Index Terms*—Confusion network combination, cross adaptation, discriminative training, GALE, hidden activation temporal patterns (HATs), Mandarin automatic speech recognition (ASR), Mandarin pronunciations, multilayer perceptron (MLP), Tandem MLP.**

I. Introduction

With China's rapid economic growth and one of the most widely spoken languages in the world, various purposes demand a highly accurate Mandarin automatic speech recognizer (ASR). This paper seeks to achieve such a goal on broadcast news (BN) and broadcast conversational (BC) speech. We demonstrate that the core technologies developed for English ASR are applicable to a new language such as Mandarin, including discriminative acoustic model (AM) training, discriminative features, and multiple-pass unsupervised cross adaptation. However, to achieve the best performance, one needs to add language-dependent components, which for Mandarin includes extraction of tone-related features, lexical word segmentation, a tonal phone set, and optimization of automatic acoustic segmentation. We have published extraction of tone-related features in [1]–[3]. This paper will elaborate our design philosophy of word segmentation, phonetic pronunciation and acoustic segmentation. For better leveraging the core technology in cross adaptation and system combination, we further design different phone sets in component models as well as utilizing different combinations of discriminative techniques. Finally, topic-based language model (LM) adaptation with a context-based decay inference is applied before system combination.

This paper starts with a description of the acoustic and language model training data used in building the system, and the lexical word segmentation algorithm. Then, we summarize our decoding architecture, which illustrates the need for two complementary subsystems; in this setup, one can also clearly see where LM adaptation fits. The next three sections describe the key developments in the system. Section IV elaborates the improvement in automatic segmentation of long recordings into utterances of a few seconds. Section V describes the complementary subsystem design with two phone sets, two front-end features and different discriminative learning methods. Section VI describes the topic-based LM adaptation algorithm. Next, Section VII demonstrates the relative improvements of various components via experiments, and presents our 2007 evaluation result. Finally, in Section VIII, we summarize the contributions and discuss future work.

II. Speech and Text Corpora

A. Acoustic Data

In this paper, we use about 866 hours of BN and BC speech data collected by LDC for training our acoustic models, as shown in Table I. The TDT4 data do not have manual transcriptions associated with them, only closed captions. We use a flexible alignment algorithm to filter out bad segments where

Manuscript received May 12, 2008; revised November 22, 2008. Current version published July 17, 2009. This work is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract HR0011-06-C-0023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gerhard Rigoll.

M.-Y. Hwang is with Microsoft Corporation, Redmond, WA 98052-7329 USA, and also with the University of Washington, Seattle, WA 98195 USA (e-mail: mhwang@ee.washington.edu).

G. Peng was with the University of Washington, Seattle, WA 98195 USA. He is now with the Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: gpeng@ee.washington.edu).

M. Ostendorf is with Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: mo@ee.washington.edu).

W. Wang is with SRI International, Menlo Park, CA 94025 USA (e-mail: wwang@speech.sri.com).

A. Faria is with the International Computer Science Institute (ICSI), University of California at Berkeley, Berkeley, CA 94704 USA (e-mail: arlo@icsi.berkeley.edu).

A. Heidel is with National Taiwan University, Taipei 10617, Taiwan (e-mail: aaron@speech.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TASL.2009.2014263

TABLE I
MANDARIN ACOUSTIC TRAINING DATA

Corpus	Year	Duration
Hub4	1997	30 hrs
TDT4	2000-2001	89 hrs
GALE P1	2004-10/2006	747 hrs
Total	1997-2006	866 hrs

TABLE II
SUMMARY OF ALL TEST SETS. THE FIRST THREE ARE USED FOR SYSTEM DEVELOPMENT. THE LAST ONE IS THE EVALUATION SET FOR THE FINAL SYSTEM

Data	Year/Month	#Shows	BC	BN
Eval04	2004/04	3	0	1 hr
Eval06	2006/02	24	1 hr	1.16 hrs
Dev07	2006/11	74	1.5 hrs	1 hr
Eval07	2006/11,12	83	1 hr	1.25 hrs

the search paths differ significantly from the closed captions [4]. After the filtering, we keep 89 h of data for training.

As shown in Table II, we use three different test sets for system development: the EARS RT-04 evaluation set (Eval04), GALE 2006 evaluation set (Eval06), and GALE 2007 development set (Dev07).[1] Once the system parameters are finalized based on these development test sets, we then apply the settings to the GALE 2007 evaluation set (Eval07).

B. Text Data, Word Segmentation, and Lexicon

Our text corpora come from a wider range of data. In addition to those transcripts of the acoustic training data, we add the LDC Chinese Gigaword corpus, all GALE-related Chinese web text releases dated before 11/1/2006, web text downloaded and released by both National Taiwan University and Cambridge University, and the Mandarin conversational LM training data described in [5]. Word fragments, laughter, and background noise transcriptions are mapped to a special garbage word.

Like many other systems, our Mandarin ASR system is based on "word" recognition with phone-based subword units rather than character-based recognition. Word-based ASR has the advantages over character-based ASR that longer units lead to less acoustic confusability and longer character context in the language model. The potential disadvantage of word units is the possibility of a higher out-of-vocabulary (OOV) rate. We ameliorate that problem by adding single-character words for all the characters occurring in our training data.

To define words, one needs to insert spaces between sequences of Chinese characters. The SIGHAN workshop (http://www.sighan.org) aims at the optimization of Chinese word segmentation, among others. The solution can be as complicated as parsing the sentence syntax and/or semantics. However, the definition of a word in Chinese is ambiguous. Moreover, depending on the applications, the most semantically correct segmentation may not be critical. For example, in dictation applications, it is not important to realize that XYZ is a name and therefore should not be split, so long as the three characters, X, Y, Z, are recognized. Furthermore, adding all names into the decoding lexicon may not be effective for rare names due to low n-gram counts. When such names are important, postprocessing with a name recognition module is often a better approach. Based on this reasoning and with the goal of minimizing computational cost, we avoid the route of building a Chinese parser, but seek an algorithm that is fast, statistically robust, and consistent with n-gram language modeling. Hence, the n-gram-based word segmenter arises naturally.

Starting from the 64K-word BBN-modified LDC Chinese word lexicon, we manually augment it with 20K new words (both Chinese and English words) over time from various sources of frequent names and word lists. We then apply a simple longest-first match (LFM) algorithm with this 80K-word list to segment all training text. The search lexicon is then given by the most frequent 60 K words, together with single-character words for all those characters that occur in the training data. We do not add other unseen single characters into the lexicon. It is a tradeoff between covering all possible character sequences (zero Chinese OOV), and minimizing the acoustic confusability and LM perplexity. With this lexicon, we train an n-gram LM, with all OOV words mapped to the garbage word. In total, after word segmentation, the LM training corpora comprise around 1.4 billion words. Among them, 11 million words are from the BC genre.

LFM segmentation is simple and fast. However, LFM does not take context into account and sometimes makes inappropriate segmentation errors that result in wrong or difficult-to-interpret semantics, as the following example shows:

(English) The Green Party and Qin-Min Party reached a consensus.

(LFM) 民进党 和亲 民党 达成 共识 (wrong)

(1gram) 民进党 和 亲民党 达成 共识 (correct)

To improve word segmentation, we have been advocating maximum-likelihood (ML) search based on an n-gram LM. A lower-order n-gram is preferred if the LM is ML-trained on the same text, so that there is a better chance to get out of the locally optimal segmentation. Here, we use the unigram LM, trained on LFM-segmented text, to resegment the same training data. In our experience, ML word segmentation results in only slightly better perplexity and usually translates to no further improvement in recognition, possibly because with the complexity of our system, a minor improvement in word segmentation does not yield noticeable impact.[2]

After the ML word segmentation, we then retrain our n-gram LMs, smoothed with the modified Kneser–Ney algorithm [6] using the SRILM toolkit [7]. N-grams of the garbage word are also trained. Table III lists the sizes of the full n-gram LMs and their pruned versions. These LMs are all trained with the same text data, but with different frequency cutoffs and pruning thresholds. An n-gram is removed if its removal causes (training set) perplexity of the model to increase by less than the given threshold relatively [8]. LM_3 is pruned more than LM_4 because we apply the trigram in full search and therefore would like to

[1]The Dev07 set used here is the IBM-modified version, not the original LDC-released version.

[2]Following our algorithm, Y-C Pan at National Taiwan University indicated that their system character accuracy improved from 74.42% to 75.01%, using LFM vs. ML segmentation.

TABLE III
NUMBERS OF ENTRIES AND PERPLEXITIES IN N-GRAM LMS. qLM_n ARE THE HIGHLY PRUNED VERSIONS OF THE FULL N-GRAMS LM_n

	#2grams	#3grams	#4grams	Perplexity (Dev07-i)
qLM_3	6M	3M	—	379.8
qLM_4	19M	24M	6M	331.2
LM_3	38M	108M	—	325.7
LM_4	58M	316M	201M	297.8

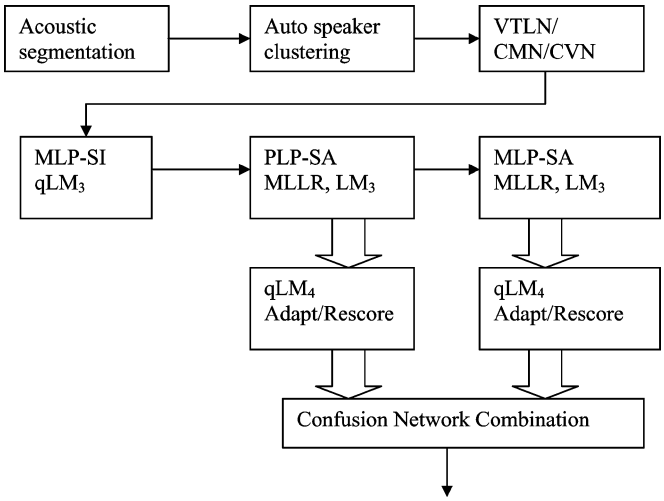

Fig. 1. System decoding architecture. Block arrows represent N-best hypotheses.

somewhat limit the search space in order to speed up the decoding process. The 4-gram, on the other hand, is applied in N-best rescoring, where we can afford a more detailed LM. The pruned n-grams (qLM_n) are further trimmed aggressively to be used in fast decoding.

To estimate the difficulty of the ASR task, we compute perplexity on a subset of Dev07 where there are no OOV words with respect to the 60 K-lexicon, ignoring any noises labeled in the reference. All noises labeled in the reference are removed when computing perplexity. This Dev07-i set contains about 44 K Chinese characters, accounting for 99% of the full Dev07 set.

III. DECODING ARCHITECTURE

Fig. 1 illustrates the flowchart of our recognition engine. We will explain our design philosophy and briefly describe the architecture in this section, while details will be presented in the next three sections.

A. Acoustic Segmentation and Feature Normalization

GALE test data come with per-show recordings. However, only specified segments in each recording are required to be recognized. Instead of feeding the minute-long segment into our decoder, we refine each segment into utterances of a few seconds long, separated by long pauses, and run utterance-based recognition. This allows us to run wide-beam search and keep rich search lattices per utterance.

Next, we perform automatic speaker clustering using Gaussian mixture models of static Mel-frequency cepstral coefficient (MFCC) features and K-means clustering. We call these speakers *auto* speakers. Note that the number of speakers in the show is unknown. Therefore, we empirically set a minimum number of utterances per speaker, tuned on the development sets. Vocal tract length normalization (VTLN) is then performed for each auto speaker, followed by utterance-based cepstral mean normalization (CMN) and cepstral variance normalization (CVN) on all features. Speaker boundary information is important, because we desire to apply speaker adaptation in later steps to improve recognition iteratively.

B. Search With Trigrams and Cross Adaptation

It is well known that speaker adaptation reduces recognition errors effectively. However, to avoid trapping in one system's own mistakes in unsupervised adaptation, it is more beneficial to use a different system's output to adapt the current system; hence, cross adaptation is used by many state-of-the-art research systems. Finally, system combination has been proved to reduce error rates successfully. Therefore, our system is composed of three recognition passes to iteratively improve itself, as shown in Fig. 1:

1) *MLP-SI Search*: The goal of the first pass is to quickly obtain a good initial transcription for adaptation. For speed reasons, we use qLM_3 and speaker-independent (SI) within-word (non cross-word, nonCW) triphone acoustic model. For accuracy reasons, we use our best signal front end with multilayer perceptron (MLP) generated phoneme posterior features. This model is denoted the MLP model, which will be elaborated in Section V-A. We were the only system that extended the MLP features to Mandarin successfully.
2) *PLP-SA Search*: Next we start unsupervised AM adaptation: use the above hypothesis to learn a speaker-dependent feature transform via speaker-adaptive training (SAT) [9], [10] and MLLR adaptation [10], on a more complicated model (CW triphones) with a different signal front end (PLP).
   After the acoustic model is speaker-adapted (SA), we then run full-trigram decoding to produce an N-best list for each utterance, in preparation of further LM adaptation and system combination.
3) *MLP-SA Search*: Similar to PLP-SA *Search*, we run cross adaptation first, using the trigram hypothesis from PLP-SA *Search* to adapt the CW triphone MLP-model, followed by full-trigram decoding to produce another N-best list. This model has the same feature front end as the one at the MLP-SI step.

C. LM Adaptation and Confusion Network Combination

Our philosophy is to use the best hypothesis at any point in time to adapt as many parameters as possible. After AM is adapted in the previous two steps, it is now time for LM adaptation. We desire to adapt higher-order n-grams, but have to resort to the pruned 4-gram due to memory constraints. Finally, to achieve the best character error rate (CER), all the words in each N-best hypothesis are split into character sequences for confusion-network based system combination (CNC) [11]. The word-to-character splitting is particularly important if different

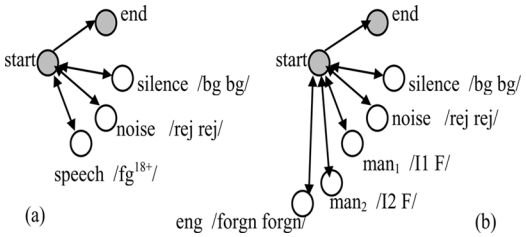

Fig. 2. Finite state grammar of our acoustic segmenter. (a) Previous segmenter. (b) New segmenter.

N-best lists are generated by different word lexicons and/or different word segmentation algorithms.

IV. Acoustic Segmentation

In error analysis of our previous system, we discover that deletion errors were particularly frequent. (Deletion errors are particularly problematic for machine translation.) Some of the deletion errors were caused by falsely recognizing speech as garbage words. To control these false alarms, we introduce a garbage penalty into the decoder, which is successful in removing some deletion errors. However, most of our deletion errors came from dropped speech segments due to faulty acoustic segmentation. Therefore, we attempt to improve acoustic segmentation so that not only fewer speech segments are dropped, but new insertion errors are simultaneously avoided [12].

A. Previous Segmenter

Our acoustic segmenter is basically a speech-silence recognizer, operated by a finite-state grammar (FSG). Fig. 2(a) shows our previous FSG. There are three "words" in the vocabulary of our previous segmenter: `silence`, `noise`, and `speech`, with "pronunciations" based on sequences of the "phones" (`bg`, `rej`, `fg`), respectively, as shown in the figure. Each pronunciation phone is modeled by a 3-state hidden Markov model (HMM), with 300 Gaussians per state. The HMMs are ML-trained on Hub4, with 39-dimensional features comprised of MFCCs and their first- and second-order differences. The segmenter operates without any knowledge of the underlying phoneme sequence contained in the speech waveform. More seriously, due to the pronunciation of `speech`, each speech segment is defined as having at least 18 consecutive `fg`s, which forces any speech segment to have a minimum duration of 540 ms, given that our front-end computes one feature vector every 10 ms.

After speech/silence is detected, segments composed of only silence and noises are discarded. Then if two consecutive speech segments are judged to be from the same speaker (based on a generalized likelihood ratio test using MFCC Gaussian-mixture models), the pause in between is less than some threshold, and the combined length is less than 9 s, then the two segments are merged.

B. New Segmenter

As all Chinese characters are monosyllabic and Mandarin is basically a CV (an optional consonant/initial followed by an obligatory vowel/final) structured spoken language, it is easy to take advantage of this language characteristic into the segmenter. In our new acoustic segmenter, we use three phonetic HMMs to model syllables with voiced initials (`I1 F`) versus syllables with voiceless initials (`I2 F`). Each of these syllables corresponds to a Chinese character. On the other hand, English words are often embedded in modern Chinese sentences. To model English speech, we allocate a separate phonetic model `forgn`. Together with silence and background noise, the input wave is thus defined as a mixed sequence of Mandarin syllables, English/foreign sounds, silence, and/or noises, as illustrated by Fig. 2(b). Meanwhile the minimum speech duration is reduced to 60 ms. Except for the finite-state grammar and the pronunciations, the rest of the segmentation process remains the same. As shown later in Section VII-A, we are able to recover most of the discarded speech segments via the new finite-state grammar and the new duration constraint.

The basic idea of increasing the model complexity and reducing the minimum duration constraint is relevant to any language, but the regular syllable structure in Mandarin makes it possible to have only a small increase in the number of HMM units (from 1 to 4 here) to model speech segments and capture some of the low energy initial sounds that may be lost by a simpler model. The new acoustic segmenter operates as efficiently as the previous one.

V. Two Acoustic Systems

As illustrated in Fig. 1, a key component of our system is cross adaptation and system combination between two complementary subsystems. Zheng [13] showed three discriminative techniques that are effective in reducing recognition errors: multilayer perceptron (MLP) features [14], minimum phone error (MPE) [15], [16] discriminative learning criterion, and featured-based MPE (fMPE) transform [17]. Table IV, cited from [13], shows that the three discriminative techniques are additive, under an SI recognition setup. However, combining all three yields minimal further improvement compared with combining only two of them, especially after unsupervised adaptation. In designing our two acoustic systems, we therefore decide to choose the most effective combinations of two techniques: MLP + MPE and fMPE + MPE, where the most effective technique, MPE training, is always applied. Furthermore, to diversify the error behaviors, we specifically design a new pronunciation phone set for the second system, with a goal of improving performance on BC speech, in particular. Except for the Hub4 training corpus, where we use the hand-labeled speaker information, we apply the same automatic speaker clustering algorithm in Fig. 1 to all other training corpora, followed by speaker-based VTLN/CMN/CVN.

All HMMs have the 3-state Bakis topology without skipping arcs. All triphone models are first ML trained, followed by MPE reestimated Gaussian means with phone lattices generated by running recognition with an unigram LM and ML-trained AM on the training data. The nonCW model, with the highly pruned trigram, is used only at Step MLP-SI for fast decoding. All later steps use CW models with SAT feature transform for the best accuracy.

All models use decision-tree-based HMM state clustering [18] to determine Gaussian parameter sharing. We settle on 3500 shared states with 128 Gaussians each, after some empirical comparison. This model size is denoted as 3500×128.

TABLE IV
ENGLISH SI WORD ERROR RATES (WER) USING THREE DISCRIMINATIVE TECHNIQUES (FROM [13])

MLP	MPE	fMPE	WER
			17.1%
yes			15.3%
	yes		14.6%
		yes	15.6%
yes	yes		**13.4%**
yes		yes	14.7%
	yes	yes	**13.9%**
yes	yes	yes	13.1%

A. System-MLP

1) HMM Observation Features: According to Table IV, the best system with two discriminative techniques is the one with MLP-feature and MPE training. Thus, our first system's front-end features consist of 74 dimensions per frame: 1) 13-dim MFCC cepstra, and its first- and second-order derivatives; 2) spline smoothed pitch feature [3], and its first- and second-order derivatives; and 3) 32-dim phoneme-posterior features generated by MLPs [14], [13].

The MLP feature is designed to provide discriminative phonetic information at the frame level. That is, each output unit represents a distinctive phone. The input layer usually covers long-span of cepstral and pitch features, to compensate the short-term cepstral features in MFCC or PLP. Its generation involves the following three main steps (in all MLPs mentioned in the paper, all hidden units use the sigmoid output function and all output units use the softmax output function):

- Generating the Tandem feature [19] by one MLP.
  We first, for each input frame, concatenate its neighboring nine frames of PLP and pitch features as the input to an MLP. Each output unit of the MLP models the likelihood of the central frame belonging to a certain phone, given the nine-frame intermediate temporal acoustic evidence. We call this vector of output probabilities the *Tandem* phoneme posteriors. The noise phone `rej` is excluded from the MLP output because it is presumably not a very discriminable class. The acoustic training data are first Viterbi aligned using an existing acoustic model, to identify the target phone label for each frame, which is then used during MLP back-propagation training.
- Generating the HATs feature [14] by a network of MLPs.
  The HATs feature is motivated from speech perception, where researchers have shown that certain frequency bands contain critical information in perceiving certain phones. To integrate that expert knowledge, we construct a two-stage network structure as illustrated in Fig. 3, where the first stage contains 19 MLPs, each fed with the log energies of a different critical band across 51 time frames, and the second stage is a single MLP, combining decisions from all critical bands to make a grand judgment.
  From [14], feeding forward the output from the *hidden* layer of the first-stage is better than feeding forward the output of the *output* layer, probably because the softmax function mitigates the discriminative power of the output layer. The output of this merger MLP is called the *HATs* (*H*idden *A*ctivation *T*emporal pattern*s*) phoneme posteriors.
- Combining Tandem and HATs.
  Finally, the 71-dim *Tandem* and *HATs* posterior vectors are combined using the Dempster–Shafer [20] algorithm. The values in the combined vector still range from 0 to 1. We then apply logarithm to make the posterior distributions appear more like a Gaussian. Principal component analysis (PCA) is then applied to the log posteriors to 1) make each dimension independent, as our HMM models use Gaussian mixtures with diagonal covariances, and 2) reduce the dimensionality from 71 to 32. The 32 dimensions of phoneme posteriors are then appended to the MFCC and pitch features. This system with 74-dim features is thus referred to System-MLP because of the use of the MLP features.

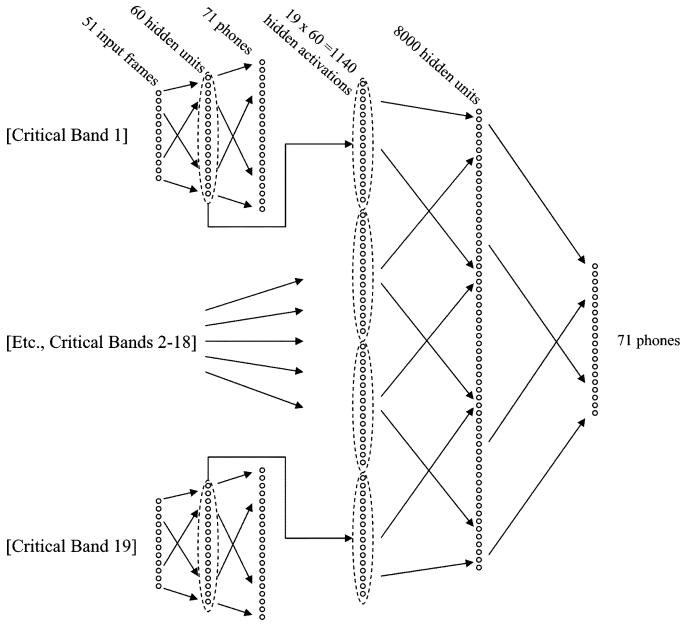

Fig. 3. *HATs* feature, computed using a two-stage MLP.

2) Pronunciation Phone Set: In this system, we inherit the BBN pronunciation dictionary with our minor fixes, and augment it with additional words discovered in Section II-B. This phone set avoids infrequent phones to reduce the size of the phone set in order to better train all parameters, after following the main vowel idea proposed in [21]. Vowels usually have four distinct tones. Neutral tones is replaced by the third tone. The rare `I2` as in 词 is modeled by `I1`. The key idea behind main vowels is to divide the Mandarin final into a glide and a main vowel; attach the glide to the initial, and allow only the main vowel to be tonal. BBN further separates the glide and the initial, and divides the main vowel into a kernel and an optional coda, allowing even a higher degree of parameter sharing, such as `eng` = `e NG` and `en` = `e N`. There are many Mandarin ASR systems using Mandarin initials and tonal finals as the basic HMM unit, which results in a large number of base phones, making triphone clustering less reliable as there are too many rare triphones [22]. Our system is currently limited to triphone

TABLE V
MAIN-VOWEL PHONETIC PRONUNCIATIONS FOR CHINESE SYLLABLES

Sample character	Pinyin	Phonetic pronunciation
抗	kang	k a2 NG
南	nan	n A2 N
要	yao	y a4 W
蛾	e	e2
奔	ben	b e1 N
绷	beng	b e1 NG
而	er	er2
耶	ye	y E1
吹	chui	ch w E1 Y
我	wo	w o3
有	you	y o3 W
医	yi	y i1
因	yin	y i1 N
不	bu	b u4
之	zhi	zh IH1
吃	chi	ch IH1
是	shi	sh IH4
日	ri	r IH4
资	zi	z I1
差	ci	c I1
斯	si	s I1
充	chong	ch o1 NG
捐	juan	j v A1 N
虚	xu	x yu1
玉	yu	v yu4
云	yun	v e2 N

modeling. If we would advance to quinphone modeling, the explosion will be even more serious. While those systems using initials and finals often resort to rule-based clustering, the authors favor the kernel-coda design with automatic parameter clustering. Table V lists some key syllables for readers' better understanding when converting Pinyin to the kernel-coda phonetic pronunciation. Notice that the phones are case sensitive.

In addition to the 70 phones, we add one phone designated for silence, and another one, `rej`, for modeling all noises, laughter, and foreign (non-Mandarin) speech. Both the silence phone and the noise phone are context-independent. The garbage word is modeled by a pronunciation graph of two or more `rej`.

B. System-PLP

To offer a different error behavior for cross adaptation, we design the second AM with a different signal front end and a different phonetic pronunciation.

1) PLP Feature and fMPE Feature Transform: Similar to other DARPA participants, the first difference in the second AM is in switching to a different cepstral feature: from MFCC to PLP. In addition, following the advice from Table IV, we apply fMPE in our second system.

To compute the fMPE feature transform, we first train a smaller (3500×32) CW ML model with SAT feature transforms. A smaller model is adopted for its computational efficiency. For each time frame, we compute all Gaussian density values of five neighboring frames, given the 3500×32 model. This gives us a Gaussian posterior vector, h_t, of $3500 \times 32 \times 5 = 560\text{K}$ dimensions. The final feature used is $z_t = (A_k x_t + b_k) + M h_t$, where x is the 42-dim PLP feature, A_k and b_k are the speaker-dependent SAT feature transform, and M is a global and sparse $42 \times 560\text{K}$ transformation matrix which is learned through an MPE criterion [17]. Finally, with the z_t feature, we train 3500×128 CW MPE models to be used at Step PLP-SA.

TABLE VI
DIFFERENCE BETWEEN THE 72-PHONE AND 81-PHONE SETS ASTERISKS INDICATE CONTEXT-INDEPENDENT PHONES

Sample	Phone-72	Phone-81
要	a W	aw
北	E Y	ey
有	o W	ow
爱	a Y	ay
安	A N	a N
次	I	i
尺	IH	i
了	e3	e5
吗	a3	a5
子	i3	i5
victory	w	V^*
呃	o3	fp_o^*
嗯	e3 N	fp_en^*

2) Pronunciation Phone Set: In designing our second acoustic system, we also attempt to offer a better modeling particularly for BC speech. Since BC speech tends to be faster and more sloppy, we first introduce a few diphthongs as shown in Table VI, where vowels with no tone apply to all four tones. Combining two phones into one reduces the minimum duration requirement by half and hence is likely better for fast speech. The addition of diphthongs also naturally removes the need for the syllable-ending `Y` and `W` sounds. Next we add three neutral-tone phones for the few highly frequent neutral-tone characters. Furthermore, we add phone `V` for the v sound in English words, as this phone is missing in Mandarin but not difficult at all for Chinese people to pronounce correctly (`w` is used in the 72-phone set to emulate `V`). Separating them makes Chinese `w` glide purer and thus more accurate. Similarly, we add two different phones for the two most common filled-pause characters (呃,嗯) to separate them from those parameters for regular Chinese words. As there is not much training data for `V` and the filled pauses, we make these three phones context-independent, indicated by the asterisks.

Finally, to keep the size of the new phone set manageable and thus the Markov state clustering reliable, we merge `A` into `a`, and both `I` and `IH` into `i`. We rely on triphone modeling to distinguish these allophones of the same phoneme. With `I2` represented by `i2`, the second tone of the nonretroflex `i` is now modeled correctly.

VI. Topic-Based Language Model Adaptation

After generating the two N-best lists for each utterance, we update the language scores using an adapted higher-order n-gram.

We perform topic-based LM adaptation using a Latent Dirichlet Allocation (LDA) topic model [23], [24]. The topic inference algorithm takes as input a weighted bag of words $\boldsymbol{w}$ (e.g., the words in a topically coherent story) and returns the topic mixture $\boldsymbol{\theta}$. The k-topic LDA model is trained on the same LM training data used for the general LMs described earlier, and is then used to further decompose the LM training data into k topic-specific text corpora. Each training sentence is labeled with the topic which has the maximum weight in the $\boldsymbol{\theta}$ derived from that sentence. We then use the resulting topic-specific corpora to train one n-gram LM per topic [25], [26], using modified Kneser–Ney smoothing.

During decoding, we infer the topic mixture weights dynamically for each utterance using the unigram LDA model. We select the top few most relevant topics above a threshold, and use their weights in $\boldsymbol{\theta}$ to interpolate with the background language model in Table III. The LDA model is used only for finding the weights, rather than as the adaptation distribution itself (as in [27], [28], where the unigram adaptation distribution is incorporated using maximum entropy), making it practical to adapt topic-specific higher-order n-grams. This can be seen as a cluster-based approximation of the topic mixture when extended from unigram $(\boldsymbol{\theta})$ to n-gram distributions (the linear interpolation of topic LMs according to the weights in $\boldsymbol{\theta}$). Because there are multiword sequences that are topic-dependent, the refined n-gram probabilities benefit adaptation, but do not add to the cost of LDA inference.

In order to make topic inference more robust against recognition errors, we weigh the words in $\boldsymbol{w}$ based on an N-best-list derived confidence measure. Additionally, we include words not only from the utterance being rescored, but also from surrounding utterances in the same story chunk via an exponential decay factor, where the words of distant utterances are given less weight than those of nearer utterances [26]. The use of weighted bags of words and of context yields dependable topic inference even for highly erroneous inferences and hence better recovery from those recognition errors in the system hypotheses. As a heuristic, utterances that are in the same show and less than 4 seconds apart are considered to be part of the same story chunk. The adapted n-gram is then used to rescore the N-best list.

VII. Experimental Results

A. Acoustic Segmentation

Tables VII and VIII show the CERs with different segmenters at the *MLP-SI Search* step and PLP-SA *Search* step, respectively, on Eval06. The error distributions and our manual error analysis both show that the main benefit of the new segmenter is in recovering lost speech segments and thus in lowering deletion errors. However, those lost speech segments are usually of lower speech quality and therefore lead to more substitution errors. For comparison, we also show the CERs with the oracle segmentation as derived from the reference transcriptions. These results show that our segmenter is very competitive.

TABLE VII
CERs at Step MLP-SI on Eval06

Segmenters	Sub	**Del**	Ins	Overall
Previous segmenter	9.7%	**7.0%**	1.9%	18.6%
New segmenter	9.9%	**6.4%**	2.0%	18.3%
Oracle segmenter	9.5%	**6.8%**	1.8%	18.1%

TABLE VIII
CERs at Step PLP-SA on Eval06

Segmenters	Sub	**Del**	Ins	Overall
Previous segmenter	9.0%	**5.4%**	2.0%	16.4%
New segmenter	9.2%	**4.8%**	2.1%	16.1%
Oracle segmenter	8.8%	**5.3%**	2.0%	16.1%

TABLE IX
SI Bigram CERs on Eval04, Using NonCW ML Models Trained on Hub4

HMM Feature	MLP Input	CER
(a) MFCC	—	24.1%
(b) MFCC+F0	—	21.4%
(c) MFCC+F0+Tandem	PLP	20.3%
(d) MFCC+F0+Tandem	PLP+F0	19.7%

B. Pitch and MLP Features

Since Mandarin is a tonal language, it is well known that adding pitch information helps with speech recognition. For this reason, we investigate adding pitch into the input of the Tandem neural nets. For quick verification, we used Hub4 to train nonCW triphone ML models. Table IX shows the SI bigram CER performance on Eval04. Pitch information (F0) obviously provides extra information for both the MFCC front-end and the *Tandem* front-end. Comparing (b) and (d), it also demonstrates the discriminative power offered by the MLP feature. When we expand the training data, the performance difference between the MLP system versus the PLP system is even more obvious (see [29, Table III]). This is because of the improved neural nets and thus improved phoneme posteriors, while cepstral-only front ends benefit only in cepstral features, which the MLP system has as well. That is, the more free parameters a system has, the more beneficial increasing training set becomes.

C. Pronunciation Phone Sets

To perform a fair comparison, two nonCW triphone PLP models are ML-trained with the 866 hours of data: one with the 81-phone set and the other with the 72-phone set. These comparisons are conducted with the SI models and the pruned trigram, as shown in Table X.

A careful error analysis reveals that the improvement in the BC portion from the 81-phone set is completely due to the reduction in deletion errors, possibly indicating the effectiveness of the diphthongs for fast speech. Therefore, despite the modest overall improvement, the new phone set achieves our goal of generating different error patterns.

TABLE X
CERs ON DEV07 USING DIFFERENT PHONE SETS. BOTH MODELS ARE NONCW PLP ML TRAINED. ONE-PASS SI DECODING IS RUN WITH qLM_3

Model	BN	BC	Avg
Phone-81	7.6%	27.3%	18.9%
Phone-72	7.4%	27.6%	19.0%

TABLE XI
NONCW SI qLM_3 DECODING AT STEP MLP-SI ON DEV07. ALL AMS ARE TRAINED ON 866 h OF SPEECH WITH THE PHONE-72 PRONUNCIATION

Training objective	Feature front end	CER
(1) MPE	MLP	14.1%
(2) ML	MLP	15.1%
(3) ML	PLP	19.0%

TABLE XII
DECODING PROGRESS ON DEV07. ALL AMS ARE TRAINED ON 866 h OF SPEECH

LM/Search	MLP-SI	PLP-SA	MLP-SA	CNC
(1) qLM_3	14.1%	–	–	–
(2) LM_3	–	12.0%	11.9%	–
(3) adapted qLM_4	–	11.7%	11.4%	11.2%
(4) LM_4	–	11.9%	11.7%	11.4%

D. Acoustic and Language Model Adaptation

Tables XI and XII shows the decoding progress following the architecture depicted in Fig. 1, on Dev07 with all 866 hours of acoustic training data. The first row of Table XI shows the CER at Step MLP-SI. The second row is to compare ML training versus MPE training. The third row confirms once again that the more training data there is, the larger improvement the MLP feature contributes, compared with the difference between Row (b) and (d) of Table IX. Notice that to make a fair comparison, the third system is trained with the same phone-72 pronunciation. However, when one compares the results at Step PLP-SA and *MLP-SA* in Row (2)–(4) of Table XII, it is interesting to notice that after MPE training, cross-reference speaker adaptation and more detailed LMs, the adapted PLP system is not much worse than the adapted MLP system. This indicates that the improvement provided by each technique overlaps each other.

Due to memory constraints, we are unable to adapt the full 4-gram LM. Instead, we train 64 topic-dependent 4-grams and interpolate them with qLM_4. The adapted 4-gram is then applied to rescore the N-best list of each utterance. The result is shown in the third row in Table XII. Compared with the full static 4-gram in the next row, the adapted 4-gram is slightly but consistently better.

E. System Combination

Finally, a character-level confusion network combination (CNC) [11] of the two rescored N-best lists yields a 11.2% CER on Dev07, as shown in the last column of Table XII. When this system was officially evaluated on Eval07 in June 2007, we achieved the best GALE Mandarin ASR error rate of 9.1% (BN 3.4%, BC 16.3%)[3] as shown in Table XIII.

TABLE XIII
OFFICIAL CERs ON EVAL07. SYSTEM PARAMETER SETTINGS WERE TUNED ON DEV07

LM/Search	MLP-SI	PLP-SA	MLP-SA	CNC
(1) qLM_3	12.4%	–	–	–
(2) LM_3	–	10.2%	9.6%	–
(3) adapted qLM_4	–	9.7%	9.3%	9.1%

VIII. CONTRIBUTION AND FUTURE WORK

Given existing technologies, our main contribution is to show how one can design, select, and combine them into a successful system, with careful design in language-specific issues.

The MLP-SI step itself is close to real-time, excluding the feature computation. On the other hand, the N-best generation is expensive, mainly due to the large full trigram search. In a practical system, one may stop at MLP-SI or PLP-SA with qLM_3, with predicted error rates between Row (1) and (2) in Table XII and XIII. If one chooses not to do adaptation due to speed constraint, it is important to use MPE training and MLP features to achieve high-quality recognition.

Anecdotal error analysis on Dev07 shows that diphthongs did help in examples such as 北大 (/b ey3 d a4/, Beijing University), and merging /A/ and /a/ was not harmful, but merging /I/ and /IH/ into /i/ seemed to cause somewhat more confusion among characters such as (是,至,地)=(shi,zhi,di). Perhaps we need to revert the last decision. Additionally, we may want to use a different word segmentation algorithm and/or a different word lexicon in the second system to offer more diverse error patterns.

The topic-based LM adaptation had small, but not quite satisfactory, improvement. Further refinement in the algorithm and in the implementation is needed to adapt the full 4-gram and obtain greater significance. Our previous study [30] showed that full re-recognition with the adapted LM offered more improvement than N-best rescoring. Yet the computation was expensive. A lattice or word graph re-search is worth investigating.

Finally, the system still has a much higher error rate on BC speech than BN. There is not much BC text available to train conversation-specific language models. Web crawling of more conversations is strongly needed. We are working along all the above directions.

ACKNOWLEDGMENT

The authors would like to thank SRI International for providing Decipher as the backbone of this study, and particularly for all of the technical support from A. Stolcke and Z. Jing.

REFERENCES

[1] M. Y. Hwang, X. Lei, T. NG, I. Bulyko, M. Ostendorf, A. Stolcke, W. Wang, and J. Zheng, "Progress on Mandarin conversational telephone speech recognition," in *Proc. Int. Symp. Chinese Spoken Lang. Process.*, 2004, pp. 1–4.

[2] X. Lei, M. Y. Hwang, and M. Ostendorf, "Incorporating tone-related MLP posteriors in feature representation for Mandarin ASR," in *Proc. Interspeech*, 2005, pp. 2981–2984.

[3]Our official result of 8.9% CER was obtained by CNC with yet another two N-best lists generated by adapting our two systems with RWTH University top-1 hypotheses.

[3] X. Lei, M. Siu, M. Y. Hwang, M. Ostendorf, and T. Lee, "Improved tone modeling for Mandarin broadcast news speech recognition," in *Proc. Interspeech*, 2006, pp. 1237–1240.
[4] A. Venkataraman, A. Stolcke, W. Wang, D. Vergyri, V. Gadde, and J. Zheng, "An efficient repair procedure for quick transcriptions," in *Proc. ICSLP*, 2004, pp. 1961–1964.
[5] T. NG, M. Ostendorf, M. Y. Hwang, M. Siu, I. Bulyko, and X. Lei, "Web data augmented language models for Mandarin conversational speech recognition," in *Proc. ICASSP*, 2005, pp. 589–592.
[6] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Comput. Sci. Group, Harvard Univ., Cambridge, MA, TR-10-98, 1998.
[7] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Proc. 7th Int. Symp. Conf. Spoken Lang. Process.*, 2002, pp. 901–904.
[8] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 270–274.
[9] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: A maximum likelihood approach to speaker normalization," in *Proc. ICASSP*, 1997, pp. 1043–1046.
[10] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, pp. 75–98, 1998.
[11] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Comput. Speech Lang.*, pp. 373–400, 2000.
[12] G. Peng, M. Y. Hwang, and M. Ostendorf, "Automatic acoustic segmentation for speech recognition on broadcast recordings," in *Proc. Interspeech*, 2007, pp. 2977–2980.
[13] J. Zheng, O. Cetin, M. Y. Hwang, X. Lei, A. Stolcke, and N. Morgan, "Combining discriminative feature, transform, and model training for large vocabulary speech recognition," in *Proc. ICASSP*, 2007, pp. 633–636.
[14] B. Chen, Q. Zhu, and N. Morgan, "Learning long-term temporal features in LVCSR using neural networks," in *Proc. ICSLP*, 2004, pp. 925–928.
[15] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, 2002, pp. 105–108.
[16] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," in *Proc. Interspeech*, 2005, pp. 2125–2128.
[17] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP*, 2005, pp. 961–964.
[18] M. Y. Hwang, X. D. Huang, and F. Alleva, "Predicting unseen triphones with senones," in *Proc. ICASSP*, 1993, pp. 311–314.
[19] N. Morgan, B. Chen, Q. Zhu, and A. Stolcke, "Trapping conversational speech: Extending trap/tandem approaches to conversational telephone speech recognition," in *Proc. ICASSP*, 2004, pp. 537–540.
[20] F. Valente and H. Hermansky, "Combination of acoustic classifiers based on dempster-shafer theroy of evidence," in *Proc. ICASSP*, 2007, pp. 1129–1132.
[21] C. J. Chen *et al.*, "New methods in continuous Mandarin speech recognition," in *Proc. Eur. Conf. Speech Commun. Technol.*, 1997, vol. 3, pp. 1543–1546.
[22] J. L. Zhou, "Microsoft internal experiments," 2001.
[23] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Uncertainty Artif. Intell.*, 1999, pp. 289–297.
[24] D. M. Blei, A. Y. NG, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, pp. 993–1022, 2003.
[25] A. Heidel, H. A. Chang, and L. S. Lee, "Language model adaptation using latent dirichlet allocation for topic inference," in *Proc. Interspeech*, 2007, pp. 2361–2364.
[26] A. Heidel and L. S. Lee, "Robust topic inference for latent semantic language model adaptation," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, 2007, pp. 177–182.
[27] Y. C. Tarn and T. Shultz, "Unsupervised language model adaptation using latent semantic marginals," in *Proc. Interspeech*, 2006, pp. 1705–1708.
[28] Y. C. Tarn and T. Shultz, "Correlated latent semantic model for unsupervised LM adaptation," in *Proc. ICASSP*, 2007, pp. 41–44.
[29] A. Faria and N. Morgan, "When a mismatch can be good: Large vocabulary speech recognition trained with idealized tandem features," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2008, pp. 1574–1577.
[30] M. Y. Hwang, W. Wang, X. Lei, J. Zheng, O. Cetin, and G. Peng, "Advances in Mandarin broadcast speech recognition," *Proc. Interspeech*, pp. 2613–2616, 2007.

Mei-Yuh Hwang (M'94) received the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA, in 1994.

She has since worked at Microsoft Corporation, Redmond, WA, and the University of Washington, Seattle, and has published numerous conference and journal papers. Her main interests lie in pattern recognition (especially speech and handwriting recognition), statistic modeling, machine translation, heuristic search, discrete math, and algorithms. She serves as a reviewer for *Computer Speech and Language* and *Speech Communication*.

Dr. Hwang serves as a reviewer for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

Gang Peng received the B.Sc. degree in mathematics and the M.Eng. degree in computer science, both from Nankai University, Tianjin, China, in 1993 and 1998, respectively, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong SAR, in 2002.

He has worked at the City University of Hong Kong (2002–2005), the Chinese University of Hong Kong (2005–2006), the University of Washington (2006–2007), the University of Hong Kong (2007–2008), and is now affiliated with the Chinese University of Hong Kong.

His research interests include automatic speech recognition, text to speech, statistical language modeling, data mining, information retrieval, human speech perception, and cognitive aspects of human speech and language processing.

Mari Ostendorf (M'85–SM'97–F'05) received a Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1985.

She has worked at BBN Laboratories (1985–1986) and Boston University (1987–1999), and is now a Professor of electrical engineering at the University of Washington, Seattle. Her research interests are in dynamic and linguistically motivated statistical models for speech and language processing. She is a former editor of *Computer, Speech, and Language*.

Prof. Ostendorf is former Editor-in-Chief of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, and she is on the IEEE Signal Processing Society Board of Governors.

Wen Wang (M'98) received the B.S. degree in electrical engineering and the M.S. degree in computer engineering from Shanghai Jiao Tong University, Shanghai, China, in 1996 and 1998, respectively, and the Ph.D. degree in computer engineering from Purdue University, West Lafayette, IN, in 2003.

She is currently a Research Engineer at the Speech Technology and Research Laboratory, SRI International, Menlo Park, CA. Her research interests are in statistical language modeling, speech recognition, natural language processing techniques and applications, and optimization. She authored and coauthored about 40 research papers and served as reviewer for over ten journals and conferences.

Dr. Wang is member of the Association for Computational Linguistics.

Arlo Faria (M'07) received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, in 2004. He is currently pursuing the Ph.D. degree in computer science at the University of California, Berkeley.

He has been a Research Assistant at the International Computer Science Institute, University of California, Berkeley, since 2003 and a Visiting Researcher at the Center for Speech Technology Research in Edinburgh since 2005. His research interests are in automatic speech recognition and parallel computation.

Aaron Heidel received the B.S. and M.S. degrees from the College of Electrical Engineering and Computer Science, National Taiwan University, Taipei, Taiwan, where he is currently pursuing the Ph.D. degree.

His research interests include statistical language modeling and speech recognition.